International Engineering Journal For Research & Development
Vol.5
Issue 4

E-ISSN NO:-2349-0721

Impact factor : 6.193

IMAGE PROCESSING APPROACH FOR EXTRACTING TABLES FROM SCANNED DOCUMENTS

[1]Aditya Kekare, [2]Atharva Gosavi, [3]Abhishek Jachak, [4]Amit Deshmane

B.E. Student, Dept. of Computer Engineering, NBN Sinhgad School of Engineering, Ambegoan, Pune- 411041, Maharashtra, India[1,2,3]

Software Architect, bizAmica Software Pvt. Ltd., Pune, Maharashtra, India[4]

aditya.kekare@gmail.com

ABSTRACT

Due to data revolution in the 21st century, processing the ever-increasing volume of documents has become essential. Most of the data in the banking, financial and administrative disciplines is still stored on physical documents. There is a great necessity to process these documents using automation. A majority of useful data in these documents is stored in the form of tables. To maintain the value of data extracted, the data from tables needs to be extracted by maintaining the tabular structure. We have used an image processing approach for extracting these tables and the data contained in them. We perform operations on scanned documents to identify rows and columns of the table. We then extract the textual data using Optical Character recognition from each cell of the table. We used this approach for extracting bordered tables and achieved more than 90% accuracy in extracting the tabular data accurately.

***Key Words*: Image Processing, Optical Character Recognition**

INTRODUCTION

There are many open-source projects available for extracting tables and text from PDFs like Tabula, Camelot and PyPDF2.[5] We propose a solution for extracting tables for scanned documents for which, these open-source tools don't give good results.

Image Processing has quickly grown as a core research area in computer science and engineering fields. Image processing is the process of performing operations on an image to gain useful information out of it. The operations performed may enhance the image in terms of its color or format. We have used OpenCV, and open-source computer vision library that allows us to perform image processing operations on images. We demonstrate the use of various image processing operations using OpenCV on scanned images to extract tables. OpenCV allows us to perform thresholding, masking, morphological operations like erode and dilate, apply hough line transform algorithm, contour detection and much more. We use these operations first identify the lines. We then repair and enhance those lines so that they are accurately represented. We then identify the boxes in the table using the extracted lines. The boxes are then sorted to give an accurately identified structure of the table.

RELATED WORK

Sebastian et al. [2] have proposed a deep-learning based model for table detection as well as structure recognition. It is called DeepDeSRT. They propose a data-driven approach by demonstrating the use of a deep-

learning model for extracting tables. They have used the Marmot dataset for training the model. Marmot dataset is the largest open-source dataset available for PDFs with tables. DeepDeSRT works on both scanned documents as well as digital PDFs. The ICDAR 2013 dataset was used for testing the model. It achieved high accuracy on the dataset as well as a private dataset. The model demonstrates transfer learning and domain adaptation.

Shubham et al. [1] have also proposed an end-to-end deep-learning model for table detection and structure recognition called TableNet. TableNet has also been trained on the Marmot and ICDAR 2013 dataset. It provides a single solution for both the tasks of detection and structure recognition by leveraging the similarities between both the tasks. They demonstrate that the performance of the model can be improved by adding semantic features. It allows the model to exhibit transfer learning.

B. Gatos et al. [3] have proposed a novel technique for horizontal and vertical line detection as well as table detection. They have used morphological operations for detecting horizontal and vertical lines. After detecting these lines, the algorithm focuses on the intersection points of all the lines and sorts them horizontally, then vertically. Then the points are used for table reconstruction and thus maintain the format and structure of the table.

S. Deivalakshmi et al. [6] propose a table detection and content extraction technique based on morphological operations and connected components and labeling. After performing image processing, the content itself is extracted from the original document, hence not causing any degradation of the document Minghao Li et al. [4] present a new image-based table detection and recognition dataset built with Word and Latex documents. It contains 417K high quality labeled tables. This dataset can prove very useful for training deep-learning models for table detection and extraction.

PROPOSED METHODOLOGY

An image can be represented in a binary form. Each pixel in the image has a value 0 or 1 based on the format of the image. If the image is in RGB format, each pixel of the image will have 3 values representing the colors red, green and blue. But for our purposes we want to convert the image into greyscale format. In this format the image pixels have values ranging between 0 and 255, where 0 is black and 255 is white. Binary thresholding is used on the image so that the image contains only two colors, white and black. For identifying horizontal and vertical lines, which will be part of the table borders, morphological operations are used. It involves creating separate kernels for horizontal and vertical lines. Erode and dilate operations are used on the image twice using the two kernels separately. The resultant images give us the horizontal and vertical lines and eliminate any other noise from the image. These images are merged to give the image containing only the lines from the original image. This image can be used to further extract boxes from the image. But the lines in the resultant image tend to be broken or non-converging. Hence, these lines need to be processed so that they are not incomplete. A separate approach was used to merge broken lines.

PROCESSING LINES

To identify the lines from the image, Hough lines transform was used. It gives us the coordinate positions of the lines present in the image. Canny edge detection is also used to improve the accuracy of the Hough lines transform. The lines obtained from Hough transform need to be fixed. We use two criteria for identifying lines which need to be merged. Euclidean distance between two lines as well as their orientation is

taken into consideration. If the lines meet a certain threshold, they are grouped, and then merged into a single line. Similarly, all the lines within the threshold distance to each other and having similar orientation, are merged into a single line. This takes care of lines which are broken into multiple parts in the image.

On the rare chance, it can happen that some apparent convergent lines are not intersecting. This can happen due to bad input image. But this problem can also be solved using a slightly different approach. The first step of the approach is to find out lines which are incomplete. This can be done by finding lines, whose end points are not intersecting any line, but a potential intersection looms near the end point. Thus, the dangling point (the non-intersecting end point of a line) also needs to be checked for potential proximity to any other line. A group of lines is created which contains such dangling points. These are the lines which need to be further extended to meet their intersecting lines. These lines are merged by first finding out the intersection point between the two lines to be merged. There can be three different type of scenarios for merging the lines. The intersection point can lie on either of the lines which needs to be merged, or it can lie on neither of them. In the first two scenarios, only one line needs to be extended, while for the third scenario, both the lines need to be extended. The lines to be extended to meet the intersection points also need to meet a minimum Euclidean distance criterion. The orientation criterion doesn't matter here. Thus, we merge the non-converging lines and we finally get the lines accurately extracted from the image.

Fig -1: Sample Document

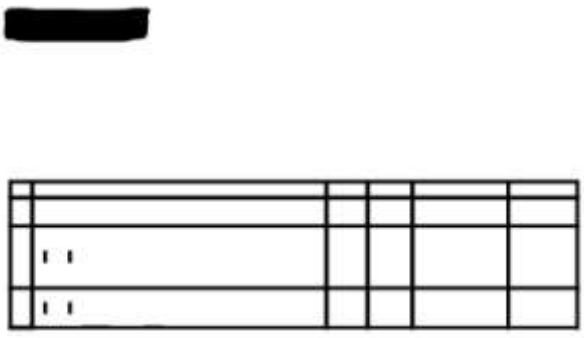

Fig -2: Line Detection in the Document

TABLE DETECTION

Contour detection is a technique for object detection in images. In our case, contour detection is used to identify boxes. Box detection gives us all the boxes contained in the image. The boxes are found based on the lines that were isolated before. Once all the boxes are identified, they are sorted top-to-bottom and left-to-right. The table can be identified using the boxes and their spatial proximity between them. The coinciding boxes will be part of the table. Thus, the boxes forming the table are extracted with their coordinates.

The next step involves extracting the text present in the boxes using Optical Character Recognition. OCR helps us to identify and extract the text in the documents accurately. The coordinates of the boxes are used to identify the text within the boxes. Thus, the table structure is identified first and then the text from the table is extracted cell-wise.

Fig -3: Table Detection with processed lines

RESULTS AND ANALYSIS

The proposed methodology was tested on more than 60 scanned documents. The documents contained tables with various formats. We achieved greater than 90% accuracy in extracting the tables correctly and accurately.

The analysis of the methodology shows that this an effective technique for extracting tables from scanned documents. The tables can be extracted accurately by applying a better threshold to the image. The kernel lengths can be changed according to different document formats to achieve better results. Thus, a better job of preprocessing the image will always yield improved results. The preprocessing can vary from image to image.

CONCLUSION

This methodology has demonstrated a valid approach towards identifying and extracting tables from scanned images. This methodology exclusively works on tables which are completely bordered. It has achieved more than 90% accuracy in extracting tables. The paper also proposes an approach for extracting lines from the image accurately. It solves the problem of broken lines in the image by correcting them, so that the table structure can be identified correctly. This approach can be used on any type of scanned document which contains data that is stored in tables.

REFERENCES

[1] Shubham Paliwal, Vishwanath D, Rohit Rahul, Monika Sharma, Lovekesh Vig, "TableNet: Deep Learning model for end-to-end table detection and tabular data extraction for scanned document images"/

https://www.researchgate.net/publication/337242893_TableNet_Deep_Learning_model_for_end-to-end_Table_detection_and_Tabular_data_extraction_from_Scanned_Document_Images

[2] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, Sheraz Ahmed, "Deepdesrt: deep-learning for detection and structure recognition of tables in document images"/
https://www.dfki.de/fileadmin/user_upload/import/9672_PID4966073.pdf

[3] Basilios Gatos, Dimitrios Danatsas, Ioannis Pratikakis, Stavros J. Perantonis, "Automatic Table detection in document images"/
https://www.researchgate.net/publication/220781373_Automatic_Table_Detection_in_Document_Images

[4] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, Zhouiun Li, "Tablebank: Table benchmark for image-based table detection and recognition"/
https://arxiv.org/abs/1903.01949

[5] Aditya Kekare, Abhishek Jachak, Atharva Gosavi, P.S. Hanwate, "Techniques for detecting and extracting tabular data from PDFs and scanned documents: A survey"/
https://www.irjet.net/archives/V7/i1/IRJET-V7I178.pdf

[6] S. Deivalakshmi, K. Chaitanya, P. Palanisamy, "Detection of table structure and content extraction from scanned documents"/
https://ieeexplore.ieee.org/abstract/document/694984